Clustering Aggregation

- References
  - A. Gionis, H. Mannila, P. Tsaparas: Clustering aggregation, ICDE 2004
  - N. Ailon, M. Charikar, A. Newman: Aggregating inconsistent information: Ranking and clustering, JACM 2008

Clustering aggregation

- Many different clusterings for the same dataset!
  - Different objective functions
  - Different algorithms
  - Different number of clusters

- How do we compare the different clusterings?

Terminology

- Clustering
  - A set of clusters output by a clustering algorithm

- Cluster
  - A group of points

Disagreement distance

- For object x and clustering C, C(x) is the index of set in the partition that contains x

- For two partitions C and P, and objects x,y in X define

$$I_{C,P}(x,y) = \begin{cases} 1 & \text{if } C(x) = C(y) \text{ and } P(x) \neq P(y) \\ & \text{OR} \\ & \text{if } C(x) \neq C(y) \text{ AND } P(x) = P(y) \\ 0 & \text{otherwise} \end{cases}$$

- if $I_{P,C}(x,y) = 1$ we say that x,y create a disagreement between partitions P and C

- $$D(P,C) = \sum_{(x,y)} I_{P,C}(x,y)$$

U	C	P
x_1	1	1
x_2	1	2
x_3	2	1
x_4	3	3
x_5	3	4

Metric property for disagreement distance

- For clustering **C**: **D(C,C) = 0**
- **D(C,C')≥0** for every pair of clusterings **C, C'**
- **D(C,C') = D(C',C)**
- Triangle inequality?
- It is sufficient to show that for each pair of points **x,y ϵX**: $I_{x,y}(C_1,C_3) \leq I_{x,y}(C_1,C_2) + I_{x,y}(C_2,C_3)$
- $I_{x,y}$ takes values 0/1; triangle inequality can only be violated when
  - $I_{x,y}(C_1,C_3)=1$ and $I_{x,y}(C_1,C_2) = 0$ and $I_{x,y}(C_2,C_3)=0$
  - Is this possible?

Which clustering is the best?

- Aggregation: we do not need to decide, but rather find a reconciliation between different groups.

The clustering-aggregation problem

- Input
  - **n** objects $X = \{x_1,x_2,\ldots,x_n\}$
  - **m** clusterings of the objects $C_1,\ldots,C_m$
    - partition: a collection of disjoint sets that cover X
- Output
  - a **single partition** C, that is as close as possible to all input partitions
-

Clustering aggregation

- Given partitions $C_1,\ldots,C_m$ find C such that

$$D(C) = \sum_{i=1}^{m} D(C, C_i)$$

the aggregation cost

is minimized

U	C_1	C_2	C_3	C
x_1	1	1	1	1
x_2	1	2	2	2
x_3	2	1	1	1
x_4	2	2	2	2
x_5	3	3	3	3
x_6	3	4	3	3

Why clustering aggregation?

- Clustering categorical data

U	*City*	*Profession*	*Nationality*
x_1	New York	Doctor	U.S.
x_2	New York	Teacher	Canada
x_3	Boston	Doctor	U.S.
x_4	Boston	Teacher	Canada
x_5	Los Angeles	Lawer	Mexican
x_6	Los Angeles	Actor	Mexican

- The two problems are equivalent

Why clustering aggregation?

- Identify the correct number of clusters
  - the optimization function does not require an explicit number of clusters

- Detect outliers
  - outliers are defined as points for which there is no consensus

Why clustering aggregation?

- Improve the robustness of clustering algorithms
  - different algorithms have different weaknesses.
  - combining them can produce a better result.

Why clustering aggregation?

- Privacy preserving clustering
  - different companies have data for the same users. They can compute an aggregate clustering without sharing the actual data.

Complexity of Clustering Aggregation

- The clustering aggregation problem is NP-hard
  - the median partition problem [Barthelemy and LeClerc 1995].
- Look for heuristics and approximate solutions.

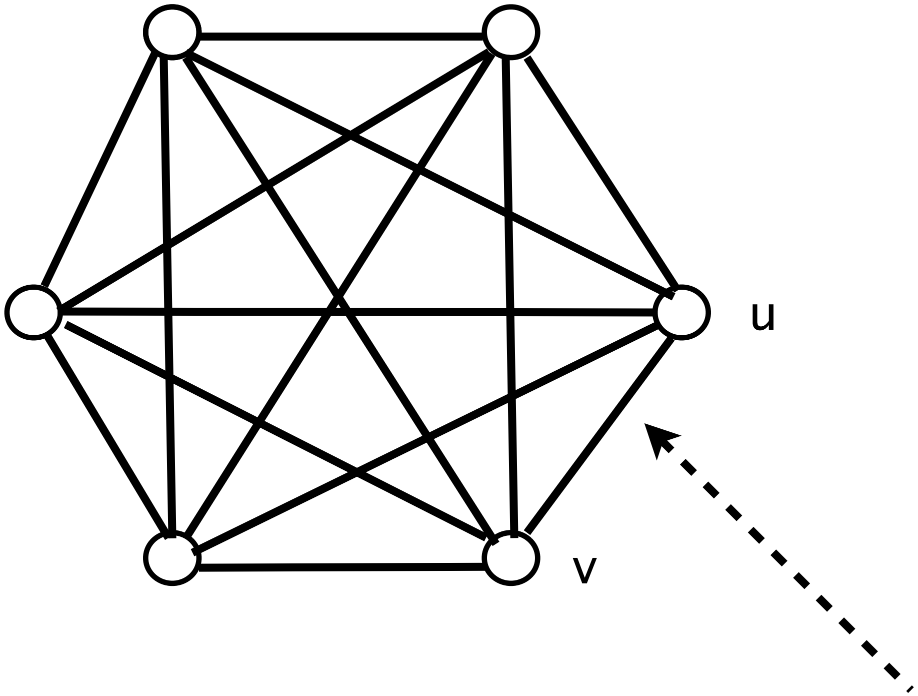
A simple 2-approximation algorithm

- The disagreement distance **D(C,P)** is a metric
- The algorithm **BEST**: Select among the input clusterings the clustering C^* that minimizes **$D(C^*)$.**
  - a 2-approximate solution. Why?

AGREEMENT graph

- The AGREEMENT graph G=(V,E) is formed as follows
  - Every node corresponds to an input point **x**
  - The weight of edge e={u,v} is the fraction of clusterings that put u and v in the same cluster

AGREEMENT graph



w(u,v): fraction of input clusterings that place u and v in the same cluster

The KwikSort algorithm

- Form the AGREEMENT graph G = (V,E)
- Start from a random node v from V
- Form cluster C(v) around v with all nodes u such that: AGREE(v,u)>=1/2
- Repeat for V = V\C(v)

A 3-approximation algorithm

- The BALLS algorithm:
  - Select a point **x** and look at the set of points **B** within distance ½ of **x**
  - If the average distance of **x** to **B** is less than ¼ then create the cluster **B∪{p}**
  - Otherwise, create a singleton cluster **{p}**
  - Repeat until all points are exhausted

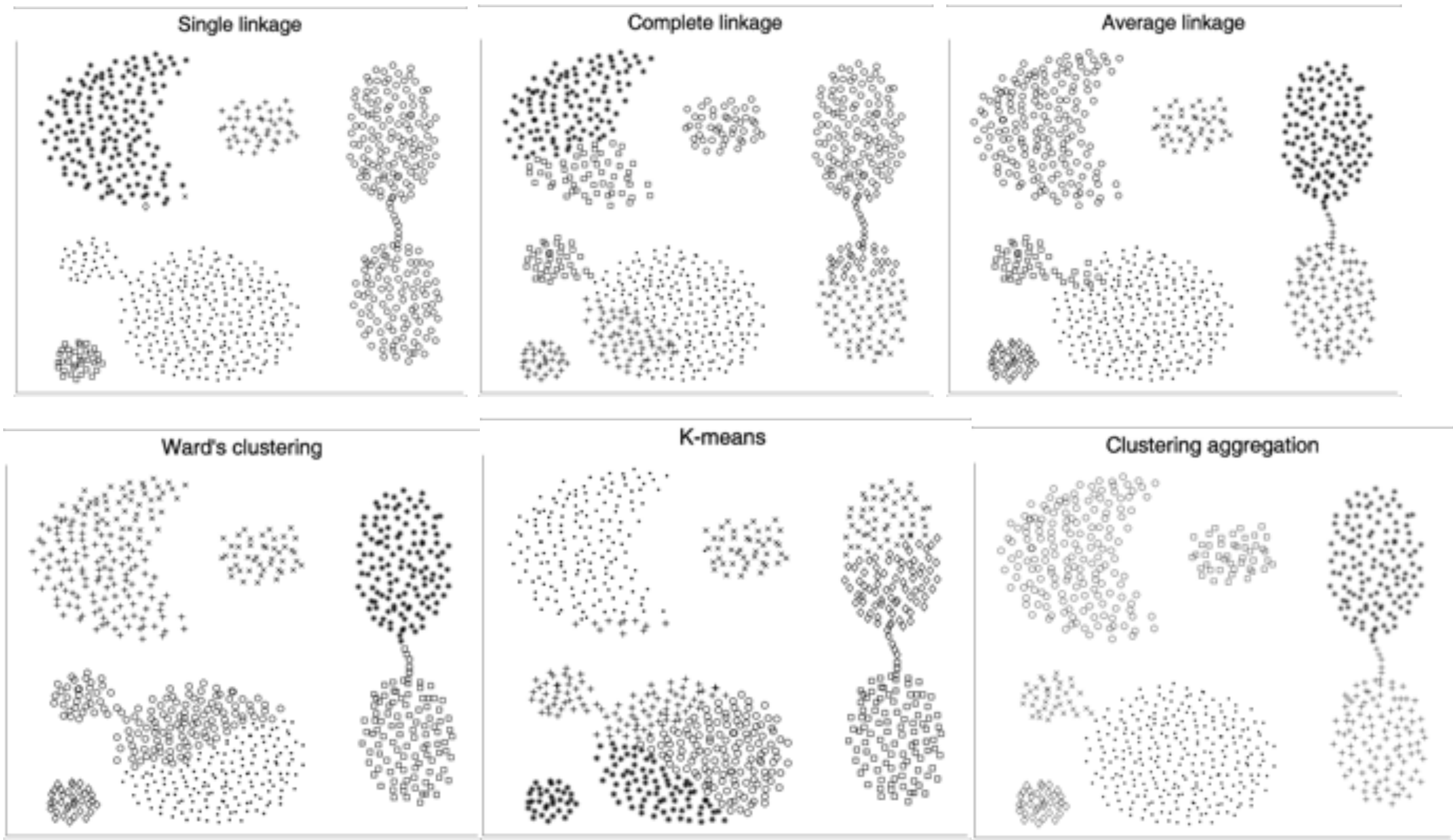
- Theorem: The **BALLS** algorithm has worst-case approximation factor **3**

Other algorithms

- **AGGLO**:
  - Start with all points in singleton clusters
  - Merge the two clusters with the smallest average inter-cluster edge weight
  - Repeat until the average weight is more than ½

- **LOCAL**:
  - Start with a random partition of the points
  - Remove a point from a cluster and try to merge it to another cluster, or create a singleton to improve the cost of aggregation.
  - Repeat until no further improvements are possible

Clustering Robustness



Clustering Robustness